WHAT IS CLAIMED IS:

1. A data processor comprising:

a processor with a predetermined instruction group as its native codes;

a hardware translator converting a code which is non-native to said processor into one or more native codes of said processor; and

a memory storing a program constituted by a native code operating on said processor,

said program stored in said memory including

a software translator program operating on said processor to convert a code that is non-native to said processor into one or more native codes of said processor and storing the converted native code in said memory,

a software interpreter program operating on said processor to sequentially interpret a code that is non-native to said processor and executing said interpreted code using a native code of said processor, and

a select program to select any of execution of a native code output by said hardware translator, execution of a native code output by said software translator and stored in said memory, and sequentially interpreting and executing non-native code by execution of said software interpreter according to a predetermined criterion to operate said processor.

2. The data processor according to claim 1, wherein said select program includes a program to select any of execution of a native code output by said hardware translator, execution of a native code output by said software translator, and sequential interpretation and execution of non-native code by execution of said software interpreter, depending upon a type and execution frequency of a read out non-native code and a status of said memory to operate said processor.

3. The data processor according to claim 1, wherein said select program includes a program to select any of execution of a native code output by said hardware translator, execution of a native code output by

said software translator, and sequential interpretation and execution of non-native code by execution of said software interpreter, depending upon a type and execution frequency of a read out non-native code and a size of an available empty region in said memory to operate said processor.

4. The data processor according to claim 1, wherein said select program includes a program to select any of execution of a native code output by said hardware translator, execution of a native code output by said software translator, and sequential interpretation and execution of non-native code by execution of said software interpreter, depending upon an address in said memory and execution frequency of a read out non-native code and an available empty region in said memory to operate said processor.

5. The data processor according to claim 1, wherein said select program includes a program to select any of execution of a native code output by said hardware translator, execution of a native code output by said software translator, and sequential interpretation and execution of non-native code by execution of said software interpreter, depending upon an address in said memory and execution frequency of a read out non-native code to operate said processor.

6. The data processor according to claim 1, wherein said select program includes a program to select any of execution of a native code output by said hardware translator, execution of a native code output by said software translator, and sequential interpretation and execution of non-native code by execution of said software interpreter every time a method constituted by said non-native code is called according to said predetermined criterion to operate said processor.

7. The data processor according to claim 1, wherein said software translator includes a code conversion program to convert non-native code into a native code so that at least a portion of a memory operand included in

said non-native code is allocated to a register provided in said processor.

8. The data processor according to claim 7, wherein said non-native code is a native code of a predetermined stack machine,

wherein said code conversion program includes a program to generate a native code effecting storage and reset of data during execution between said memory and a register of said processor so that a stack operand at a stack top side among stack operands of said memory operand is stored in a register provided in said processor.

9. The data processor according to claim 8, wherein said code conversion program further includes a program to detect non-native code that effects only data transfer to a stack and storing said detected non-native code in correspondence between a transfer source and a transfer destination thereof,

wherein said program to generate includes a program to generate a native code to effect storage and reset of data during execution between said memory and a register of said processor so that a stack operand of a stack top side from stack operands out of memory operands of said non-native code that carries out only data transfer is stored in a register provided in said processor, and for a code that uses said transfer destination as an operand, said transfer source stored in said memory is taken as an operand instead of said transfer destination.

10. An operation method of a data processor including:

a processor with a predetermined instruction group as a native code,

a hardware translator converting a code that is non-native to said processor into one or more native codes of said processor, and

a memory storing a program constituted by a native code operating on said processor,

said program stored in said memory including

a software translator program operating on said processor to convert a code that is non-native to said processor into one or more native codes of

said processor and storing said converted native code in said memory,

a software interpreter program operating on said processor to sequentially interpret a code that is non-native to said processor and executing said interpreted code using a native code of said processor,

said operation method comprising the steps of:

selecting any of execution of a native code output by said hardware translator, execution of a native code output by said software translator and stored in said memory, and sequential interpretation and execution of non-native code by execution of said software interpreter according to a predetermined criterion, and

applying to said processor for operation any of a native code output by said hardware translator, a native code output by invoking said software translator with said non-native code read out as an argument and stored in said memory, and a program code of said software interpreter with said non-native code read out as an argument, according to a selected result of said select step.

11. The operation method according to claim 10, wherein said select step comprises the step of selecting any of execution of a native code output by said hardware translator, execution of a native code output by said software translator, and sequential interpretation and execution of non-native code by execution of said software interpreter depending upon a type and execution frequency of non-native code read out and a status of said memory to operate said processor.

12. The operation method according to claim 10, wherein said select step comprises the step of selecting any of execution of a native code output by said hardware translator, execution of a native code output by said software translator, and sequential interpretation and execution of non-native code by execution of said software interpreter depending upon a type and execution frequency of non-native code read out and an available empty region in said memory to operate said processor.

13. The operation method according to claim 10, wherein said select step comprises the step of selecting any of execution of a native code output by said hardware translator, execution of a native code output by said software translator, and sequential interpretation and execution of non-native code by execution of said software interpreter depending upon an address in said memory and execution frequency of non-native code read out and an available empty region in said memory to operate said processor.

14. The operation method according to claim 10, wherein said select step comprises the step of selecting any of execution of a native code output by said hardware translator, execution of a native code output by said software translator, and sequential interpretation and execution of non-native code by execution of said software interpreter depending upon an address in said memory and execution frequency of non-native code read out to operate said processor.

15. The operation method according to claim 10, wherein said select step comprises the step of selecting any of execution of a native code output by said hardware translator, execution of a native code output by said software translator, and sequential interpretation and execution of non-native code by execution of said software interpreter every time a method constituted by said non-native code is called according to said predetermined criterion to operate said processor.

16. The operation method according to claim 10, further comprising the step of converting non-native code into a native code so that at least a portion of a memory operand included in said non-native code is allocated to a register provided in said processor by operating on said processor said software translator according to a selected result in said select step.

17. The operation method according to claim 16, wherein said non-native code is a native code of a predetermined stack machine,

wherein said code conversion step comprises the step of generating a

native code to effect storage and reset of data during execution between said memory and a register of said processor so that a stack operand at a stack top side from stack operands of said memory operand is stored in a register provided in said processor.

18. The operation method according to claim 17, wherein said code conversion step further comprises the step of detecting non-native code that effects only data transfer to a stack and storing said detected non-native code in correspondence between a transfer source and a transfer destination thereof,

wherein said generate step comprises the step of generating a native code to effect data storage and reset during execution between said memory and a register in said processor so that a stack operand at a stack top side among stack operands from a memory operand of said non-native code that effects only data transfer is stored in a register provided in said processor, and for a code that uses said transfer destination as an operand, said transfer source stored in said memory is taken as an operand instead of said transfer destination.

19. A data processor comprising:

processor means with a predetermined instruction group as a native code;

a hardware translator converting non-native code to said processor into one or more native codes of said processor;

a software translator operating on said processor to convert non-native code to said processor into one or more native codes of said processor;

storage means for storing a native code output by said software translator;

a software interpreter operating on said processor to sequentially interpret non-native code to said processor and executing said interpreted non-native code using a native code to said processor; and

selection means for selecting any of execution of a native code output by said hardware translator, execution of a native code output by said

software translator, and sequential interpretation and execution of non-native code by execution of said software interpreter to operate said processor according to a predetermined criterion.

20. The data processing apparatus according to claim 19, wherein said select means comprises means for selecting any of execution of a native code output by said hardware translator, execution of a native code output by said software translator, and sequential interpretation and execution of non-native code by execution of said software interpreter, depending upon a type or execution frequency of non-native code, or a status of said storage means to operate said processor.